(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 271 839 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.01.2003 Bulletin 2003/01

(51) Int Cl.7: H04L 9/06

AA

(21) Application number: 01310953.3

(22) Date of filing: 31.12.2001

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 28.06.2001 JP 2001195752

(71) Applicant: FUJITSU LIMITED
Kawasaki-shi, Kanagawa 211-8588 (JP)

(72) Inventors:
- Okada, Souichi, c/o FUJITSU LIMITED
Kawasaki-shi, Kanagawa 211-8588 (JP)
- Torii, Naoya, c/o FUJITSU LIMITED
Kawasaki-shi, Kanagawa 211-8588 (JP)
- Hayashi, Tomohiro,
c/o Fujitsu Comp. Techn. Ltd.
Kawasaki-shi, Kanagawa 211-8588 (JP)
- Deguchi, Chikahiro,
c/o Fujitsu Comp. Techn. Ltd.
Kawasaki-shi, Kanagawa 211-8588 (JP)
- Fujiwara, Yumi, c/o Fujitsu Comp. Techn. Ltd.
Kawasaki-shi, Kanagawa 211-8588 (JP)

(74) Representative: Hitching, Peter Matthew et al
Haseltine Lake & Co.,
Imperial House,
15-19 Kingsway
London WC2B 6UD (GB)

(54) AES Encryption circuit

(57) A round processing unit in an encryption circuit comprises: a first Round Key Addition circuit (204) that adds a round key value to input data; an intermediate register/Shift Row transformation circuit (206) that temporarily stores the output of the first Round Key Addition circuit (204) and executes Shift Row transformation; a Byte Sub transformation circuit (207) into which the values of the intermediate register/Shift Row transformation circuit (206) are inputted and which executes Byte Sub transformation; a second Round Key Addition circuit (208) into which the values of the intermediate register/Shift Row transformation circuit (206) are inputted and which adds round key values; a Mix Column transformation circuit (210) that executes Mix Column transformation upon the outputs of the second Round Key Addition circuit (208); and a second selector (203) that outputs to the second Round Key Addition circuit (204) one of the outputs of a first selector (202), the intermediate register/Shift Row transformation circuit (206), the Byte Sub transformation circuit (207), and the Mix Column transformation circuit (210). Such an encryption circuit reduces a scale of circuit and can achieve a certain level of high-speed processing in the implementation of the AES block cipher.

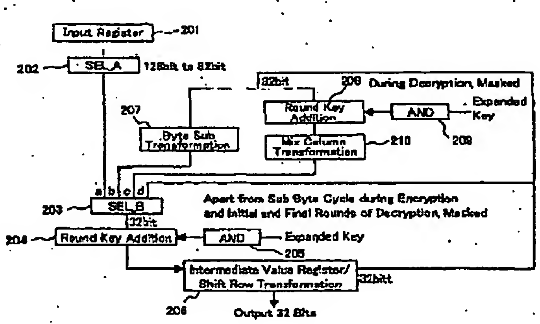

Fig. 4

EP 1 271 839 A2

Description

BACKGROUND OF THE INVENTION

Technical Field

[0001] The present invention relates to an encryption circuit for implementing in hardware the Rijndael algorithm, which is the next generation common key block encryption standard, known as the AES (advanced encryption standard), and will replace the current common.key block encryption standard in the US, called DES.

Description of Related Art

[0002] A great variety of services are being considered that involve the Internet, including electronic commerce and electronic money. These technologies are used not just in the daily lives of individuals, but also in a wide range of fields, including transactions among corporations and improving productivity. In particular, it is expected that encryption functions will be loaded onto smart cards and mobile handsets, for the purpose of verifying the identity of individuals, and that these technologies will be widely used for authentication, digital signatures, and data encryption.

[0003] Common key cryptography is used in these applications to prevent third parties from tapping on the Internet. The current standard adopted in the US for common key cryptography is DES; as its replacement, the AES (advanced encryption standard), known as the Rijndael algorithm, has been selected to be next generation common key block cryptography standard, and this algorithm is becoming the new standard. (The AES draft is available at http://csrc.nist.gov/publications/drafts/dfips-AES.pdf)

[0004] AES is a block cipher for processing in block lengths of 128 bits, and the encryption algorithm, as shown in FIG. 1, is thought to be executable by an encryption circuit comprising a round function unit 20 and a key schedule unit 10. The round function unit 20 comprises an input register 21 that temporarily stores input data, an XOR processing unit 22 that XORs the input data and expanded key segment, a round processing unit 23, a final round processing unit 24 and an output register 25 that temporarily stores output data.

[0005] The round processing unit 23 comprises a Byte Sub transformation unit 31, a Shift Row transformation unit 32, a Mix Column transformation unit 33 and a Round Key Addition unit 34; the final round processing unit 24 performs the processing of the round processing unit 23 except for the Mix Column transformation 33; it comprises a Byte Sub transformation unit 35, a Shift Row transformation unit 36 and a Round Key Addition unit 37.

[0006] Round processing iterated; the number of rounds Nr including the final round depends on the key length inputted into the key schedule unit 10, and is defined as shown in Table 1.

[Table 1]

Key Length and Number of Rounds	
Key Length	Nr
128bit	10
192bit	12
256bit	14

[0007] Thus for each key length round processing is executed Nr-1 times, and at the end the final round processing is executed. When the key length is 128 bits, round processing is executed 9 times; when 192 bits, 11 times; and when 256 bits, 13 times; and then in each case the final round processing is executed. Round keys generated at the key schedule unit 10 are inputted into the XOR processing unit 22, round processing unit 23 and final round processing unit 24.

[0008] The key schedule unit 10 generates round keys based on the key generation schedule specified in the AES draft; that algorithm is shown in FIG. 2.

[0009] The AES Proposal specification (AES Proposal: Rijndael, at http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf) introduces 2 hardware implementations for AES block cipher circuits.

[0010] One of these is a method for hardware implementation, in 128 bit units, of all the functions shown in FIG. 1 as they are (hereinafter, "conventional example 1"). In this case, for encryption and decryption, the order of processing of the functions is reversed, and thus it is necessary to prepare separate processing circuits for encryption and decryption.

[0011] Also, because, as shown in Table 1, it is necessary to change the number of times round processing is exe-

cuted depending upon the key length, it is necessary to create circuits for each key length.

[0012] Furthermore, because of the reversal of order between encryption and decryption, the order of key generation in the key schedule unit 10 for the round keys used in the round function unit 20 has to be reversed between encryption and decryption. Therefore, either there has to be 2 separate key schedule units, for encryption and for decryption, or a method has to be devised for using the key schedule unit 10 for both encryption and decryption.

[0013] The second method, as shown in FIG. 3, involves creating a coprocessor 50 that has a Byte Sub transformation unit 51 and a Mix Column transformation unit 52, and implementing in hardware only the Byte Sub transformation and the Mix Column transformation functions, and having all other functions incorporated as software into a program 41, and then processing with a CPU 40 (hereinafter, "conventional example 2").

[0014] In this case, Byte Sub transformation and Mix Column transformation, which are unsuited for processing by the CPU 40 for reasons of processing time, are implemented in hardware as the coprocessor 50, and the other processing is processed by the program 41 stored in the CPU, thus allowing the circuit scale to be reduced.

[0015] If we suppose that the AES block cipher is to be incorporated into a smart card or the like, the functions required of an encryption circuit would be to maintain a certain level of processing speed, while keeping the scale of the circuit small. With these requirements, the conventionally proposed method of implementing all the functions in 128-bit units results in the scale of circuit being too large, making the loading thereof onto a smart card difficult. With the method of implementing in hardware only the Byte Sub transformation and the Mix Column transformation, and processing the other functions with software, there is the problem of the processing speed requirements not being fulfilled.

[0016] Moreover, with the key schedule unit 10 that generates the round keys, if all the round keys are stored in memory, a large-capacity memory is needed, and this would make the scale of circuit large. Therefore, in order to reduce the scale of circuit without reducing processing speed, it is desirable to generate round keys with a circuit constitution that does not require storing the entire expanded key in memory.

SUMMARY OF THE INVENTION

[0017] It is an object of the present invention to present an encryption circuit that is small in scale and that can achieve a certain level of processing speed when implementing the AES block cipher.

[0018] The present invention provides an encryption circuit that generates from a cipher key a plurality of round keys having a number of bits corresponding to a predetermined processing block length and executing, for each processing block length, input data and round key encryption/decryption processing, by means of a round function unit comprising an XOR operation unit that XORs the input data and one of the round keys and a round processing unit that iterates round processing that includes Byte Sub transformation, Shift Row transformation, Mix Column transformation and Round Key Addition, wherein:

the round processing unit comprises: a first selector that segments input data into execution block lengths smaller than the processing block length; a first Round Key Addition circuit that adds the round key value to input data for each the execution block length; an intermediate register/Shift Row transformation circuit that temporarily stores the output of the first Round Key Addition circuit and executes Shift Row transformation using the processing block length; a Byte Sub transformation circuit wherein the intermediate register/Shift Row transformation circuit value is inputted for each the execution block length and Byte Sub transformation is executed; a second Round Key Addition circuit wherein the intermediate register/Shift Row transformation circuit value is inputted for each the execution block length and the round key value is added for each the execution block length; a Mix Column transformation circuit executing Mix Column transformation on the output of the second Round Key Addition circuit; and a second selector that outputs to the first Round Key Addition circuit one output from among the outputs of the first selector, intermediate register/Shift Row transformation circuit, Byte Sub transformation circuit, or Mix Column transformation circuit.

[0019] Here, the execution block length can be a multiple of 8 bits, the processing block length can be 128 bits and the execution block length can be 32 bits.

[0020] Further, the key length of the cipher key can be any of 128 bits, 192 bits or 256 bits.

[0021] Also, the Byte Sub transformation circuit can comprise a matrix operation unit for decryption that executes a matrix operation on input data; a third selector that outputs either the input data or the output of the matrix operation unit for decryption; an inverse operation unit for executing an inverse operation on the data outputted from the third selector; a matrix operation unit for encryption that executes a matrix operation on the data outputted from the inverse operation unit; and a fourth selector that outputs either the output of the inverse operation unit or the output of the matrix operation unit for encryption.

[0022] Further, the matrix operation unit for decryption and the matrix operation unit for encryption comprises an XOR circuit so as to perform 8-bit operations at one clock cycle and the matrix operation unit for decryption and the matrix operation unit for encryption comprises an XOR circuit so as to perform 1-bit operations at one clock cycle.

[0023] Also, the intermediate register/Shift Row transformation circuit can be used for both encryption and decryption

through the reversal of order of input of shift data relating to amount of shift for data to be inputted into the intermediate register/Shift Row transformation circuit, the input order for decryption being the reverse of the order for encryption.

[0024] Further, the Mix Column transformation circuit can comprise a plurality of multiplication units with unique multipliers and an XOR circuit that performs XOR operations for the plurality of multiplication units, the Mix Column transformation circuit executing a matrix operation between data inputted into each multiplication unit and the multiplier established for each multiplication unit. In this case, the Mix Column transformation circuit comprises 4 operation units having 4 multiplication units capable of 8-bit unit operations and XOR circuits that execute XOR operations based on the outputs of the 4 multiplication units. This multiplication units can control 2 multipliers and are used for both encryption and decryption and the multiplication units can be constituted to control addition values from high-order bits.

[0025] Also, an encryption circuit can be constituted so as to have a key expansion schedule circuit that generates from the cipher key, as an expanded key segmented into bit numbers corresponding to the execution block length, a plurality of round keys with bit numbers corresponding to a predetermined processing block length. The key expansion schedule circuit comprises:

a fifth selector that segments a cipher key into the number of bits corresponding to the execution block length and outputs the same;

a shift register to which flip-flop circuits are connected at a plurality of stages, the flip-flop circuits latching data in units of the execution block length;

a first XOR circuit that XORs the output of the final stage flip-flop circuit of the shift register with one constant selected from among a group of constants;

a sixth selector into which are inputted the outputs of those flip-flops of the shift register that are involved in operations for encryption and the outputs of those flip-flops involved in operations for decryption, and which selectively outputs one of these;

a Rot Byte processing circuit that rotates the output of the sixth selector;

a seventh selector into which the output of the sixth selector and the output of the Rot Byte circuit is inputted and which selectively outputs one of these;

a Sub Byte processing circuit that executes Byte Sub transformation on the output of the seventh selector for each the execution block length;

an eighth selector into which the output of the sixth selector and the output of the Sub Byte processing circuit are inputted, and which selectively outputs one of these;

a second XOR circuit that executes an XOR operation based on the output of the first XOR circuit and the output of the eighth selector; and

a shift register unit selector that selectively outputs, to those flip-flops of the shift register the outputs of which are subject to operations for encryption, either the output of the second XOR circuit or the output of the adjacent stage flip-flop.

[0026] Here, the shift register comprises 8 flip-flops executing data processing in 32-bit units, and the sixth selector is constituted so that the outputs of the second, fourth, sixth and eighth flip-flops from the bottom from among the flip-flops are inputted therein, and that it outputs one of these.

[0027] Also, through the input into the seventh selector of the output of the intermediate register/Shift Row transformation circuit and the input into the second selector of the output of the Sub Byte processing circuit, a single circuit can be used for the Sub Byte processing circuit and the Byte Sub transformation circuit of the round processing unit.

[0028] From the following detailed description in conjunction with the accompanying drawings, the foregoing and other objects, features, aspects and advantages of the present invention will become readily apparent to those skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

[0029]

FIG. 1 is a block diagram of AES processing using the Rijndael algorithm;

FIG. 2 is a key schedule program list;

FIG. 3 is a block diagram showing one envisioned circuit implementation;

FIG. 4 is a block diagram of a round function unit adopted in a first embodiment of the present invention;

FIG. 5 is a block diagram showing an intermediate register/Shift Row transformation circuit;

FIG. 6 is a block diagram showing a Mix Column transformation circuit;

FIG. 7 is a block diagram showing the constitution of a multiplication unit;

FIG. 8 is a block diagram showing another constitution of a multiplication unit;

FIG. 9 is a block diagram showing a key schedule unit;
FIG. 10 is a block diagram showing a Byte Sub transformation circuit;
FIG. 11 is a block diagram showing a matrix operation circuit for encryption;
FIG. 12 is a block diagram showing a matrix operation circuit for decryption;
FIG. 13 is a block diagram showing another example of a matrix operation circuit for encryption; and
FIG. 14 is a block diagram showing another example of a matrix operation circuit for decryption.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Round Function Unit

[0030] The AES block cipher is an algorithm that encrypts/decrypts the 128 bit data with the 128 bit, 192 bit or 256 bit key. As shown in FIG. 1, it comprises a key schedule unit 10 that generates a plurality of round keys from the cipher key, and a round function unit 20 that uses the round keys inputted from the key schedule unit 10 to encrypt and decrypt. The round function unit 20 performs such processing as XOR operations, Byte Sub transformation processing, Shift Row transformation processing, Mix Column transformation processing, Round Key Addition processing.

[0031] The first embodiment of the present invention is a circuit for implementation of this round function unit 20, and the constitution of this circuit is shown in FIG. 4. Each circuit block executes 32-bit processing with the exception of Shift Row transformation processing, which is 128-bit processing; transfer of data between circuit blocks is executed in 32-bit units.

[0032] This round function unit contains: an input register 201 that temporarily stores input data; a first selector 202 that selects 32-bit data from the 128-bit input data; a second selector 203 into one input terminal of which the output of the first selector 202 is inputted; a first Round Key Addition circuit 204 into which the output of the second selector 203 is inputted; an add data selector 205 that inputs into the first Round Key Addition circuit 204 an expanded key segment or "0"; an intermediate register/Shift Row transformation circuit 206 that stores the output value of the first Round Key Addition circuit 204 and executes Shift Row transformation in 128-bit units; a Byte Sub transformation circuit 207 into which intermediate register/Shift Row transformation circuit 206 values are inputted and which executes Byte Sub transformation; a second Round Key Addition circuit 208 into which intermediate register/Shift Row transformation circuit 206 values are inputted for each 32 bits; an add data selector 209 which inputs into the second Round Key Addition circuit 208 an expanded key segment or "0"; and a Mix Column transformation circuit 210 which executes Mix Column transformation on the output of the second Round Key Addition circuit 208. The outputs of the first selector 202, Byte Sub transformation circuit 207, Mix Column transformation circuit 210, and intermediate register/Shift Row transformation circuit 206 are inputted into the second selector 203, and one of these outputs is outputted to the first Round Key Addition circuit 204.

Operation Schedule during Encryption

[0033] The operation schedule during encryption in the round function unit is shown in Table 2.

[Table 2] Round Function Operation Schedule

Round	Cycle	Processing	SEL_B
0	000–003	Round Key Addition	a
1	004–007	Byte Sub Transformation	b
	008	Shift Row Transformation	c
	009–012	Mix Column Transformation Round Key Addition	c
2	013–016	Byte Sub Transformation	b
	017	Shift Row transformation	c
	018–021	Mix Column Transformation Round Key Addition	c
	Omitted		
Nr–1	#1	Byte Sub Transformation	b
	(Nr–1)*9–1	Shift Row Transformation	c
	(Nr–1)*9 – (Nr–1)*9+3	Mix Column Transformation Round Key Addition	c
Nr	#2	Byte Sub Transformation	b
	Nr*9–1	Shift Row Transformation	d
	Nr*9 – Nr*9+3	Round Key Addition	d

#1 : (Nr–1)*9–5 – (Nr–1)*9–2
#2 : Nr*9–5 – Nr*9–2

Note: The table shows operations during encryption. In decryption, the order of round key and Mix Column processings is switched.

[0034] Here, in round 0, addition of an expanded key segment is executed by the first Round Key Addition circuit 204 with a selector position of "a" for the second selector 203. Input data in the input register 201 is selected in 32 bit units by the first selector 202 and inputted into the first Round Key Addition circuit 204, and to this is added a portion of a round key, inputted from the key schedule unit, this portion being a 32-bit segment of the expanded key. While the input data and the expanded key are being changed into 32-bit units, the first Round Key Addition circuit 204 executes addition processing, and the XOR processing of the XOR unit 22 in FIG. 1 is thereby executed on 128-bit processing blocks in the 4 cycles of cycles 000 through 003. The result of the operation by the first Round Key Addition circuit 204 is stored in order in 32-bit units in the intermediate register/Shift Row transformation circuit 206.

[0035] In round 1, the round processing 23 in FIG. 1 is executed, and Byte Sub transformation processing 31, Shift Row transformation processing 32, Mix Column transformation processing 33, and Round Key Addition processing 34 are executed. Thus, first of all, in cycles 004 through 007, with a selector position of "b" for the second selector 203,

the data stored in the intermediate register/Shift Row transformation circuit 206, while being shifted in 32-bit units, is read out and inputted into the Byte Sub transformation circuit 207. At this time, by making the data to be selected by the add data selector 205 "0", the first Round Key Addition circuit 204 is put into a masked state. The result of the operations of Byte Sub transformation circuit 207 is stored in order in 32-bit units in the intermediate register/Shift Row transformation circuit 206. Thus Byte Sub transformation processing performs on 128 bits, and the result is stored in the intermediate register/Shift Row transformation circuit 206.

[0036] Next, in cycle 008, Shift Row transformation processing is executed. The intermediate register/Shift Row transformation circuit 206 is capable of executing Shift Row transformation processing in 128-bit units, and in this cycle 008, 128-bit Shift Row transformation processing is executed. At this time, the selector position of the second selector 203 may be any position, but in consideration of the processing in the next cycle, a position of "c" is preferable.

[0037] In cycles 009 through 0012, Mix Column transformation processing and Round Key Addition processing are executed. Herein, the data stored in the intermediate register/Shift Row transformation circuit 206, while being shifted in 32-bit units, is read out and inputted into the second Round Key Addition circuit 208. At this time, by making the data to be selected by the add data selector 209 "0", the second Round Key Addition circuit 208 is put into a masked state. By setting the selector position of the second selector 203 at "c", the data upon which Mix Column transformation processing has been executed at the Mix Column transformation circuit 210 is inputted into the first Round Key Addition circuit 204 via the second selector 203. An expanded key segment to be inputted from the key schedule unit is selected for data to be selected by the add data selector 205, and this data undergoes Round Key Addition processing at the first Round Key Addition circuit 204. The result of the Mix Column transformation processing at the Mix Column transformation circuit 210 and the Round Key Addition processing at the first Round Key Addition circuit 204 are, while being each shifted in 32-bit units, stored in the intermediate register/Shift Row transformation circuit 206. Thus, the result of the 128 bits upon which Mix Column transformation processing and the Round Key Addition processing were executed in cycles 009 through 012 are stored in the intermediate register/Shift Row transformation circuit 206. In this manner, one round of processing is executed in the 9 cycles of cycles 004 through 012.

[0038] Next, in rounds 2 through (Nr-1), the same processing as in round 1 is executed (however, Nr is the number of processing rounds including the final round, and as shown in Table 1, the number of rounds will differ according to key length).

[0039] In round Nr (the final round), the final round processing 24 of FIG. 1 is executed; this comprises Byte Sub transformation processing 35, Shift Row transformation processing 36 and Round Key Addition processing 37.

[0040] Thus in cycles (Nr*9-5) through (Nr*9-2), with the selector position of the second selector 203 at "b", data stored in the intermediate register/Shift Row transformation circuit 206, while being shifted in 32-bit units, is read out and inputted into the Byte Sub transformation circuit 207. At this time, by making the data to be selected by the add data selector 205 "0", the first Round Key Addition circuit 204 is put into a masked state. The result of the operation of the Byte Sub transformation circuit 207 is stored in order in 32-bit units in the intermediate register/Shift Row transformation circuit 206. Thus Byte Sub transformation processing of 128 bits is performed, and the result is stored in the intermediate register/Shift Row transformation circuit 206.

[0041] Next, in the (Nr*9-1) cycle, 128-bit Shift Row processing is executed. At this time, the selection position of the second selector 203 may be any position, but in consideration of the processing of the next cycle, a position of "d" is preferable.

[0042] In the (Nr*9) through (Nr*9+3) cycles, Round Key Addition processing is executed. Specifically, by making the selector position of the second selector 203 "d", the data stored in the intermediate register/Shift Row transformation circuit 206, while being shifted in 32-bit units, is read out and inputted into the first Round Key Addition circuit 204 via the second selector 203. At this time, by making data to be selected by the add data selector 205 an expanded key segment to be inputted from the key schedule unit, the first Round Key Addition circuit 204 adds 32-bit round keys. The result of the Round Key Addition processing by the first Round Key Addition circuit 204 is stored in the intermediate register/Shift Row transformation circuit 206 while being shifted in 32-bit units. Thus in the (Nr*9) through (Nr*9+3) cycles, the result of the Round Key Addition processing on the 128 bits is stored in the intermediate register/Shift Row transformation circuit 206. In this manner, in the 9 cycles from (Nr*9-5) through (Nr*9+3), final round processing is executed.

Operation Schedule during Decryption

[0043] Operations during decryption in this round function unit are performed in the reverse order to operations during encryption. This operation schedule is shown in Table 3.